#include<stdio.h>

int visited[10],n,cost[10],l=0;

static int node\_no=1;

int a[10][10];

int find\_row\_min(int b[][10],int r)

{

int min=999,j;

for(j=0;j<n;j++)

{

if(b[r][j]<min)

min= b[r][j];

}

if(min==99)

min=0;

return min;

}

int find\_col\_min(int b[][10],int c)

{

int i,min=999;

for(i=0;i<n;i++)

{

if(b[i][c]<min)

min= b[i][c];

}

if(min==99)

min=0;

return min;

}

int min1(int temp[][2],int q)

{

int i,j,small=999;

for(i=0;i<q;i++)

{

if(temp[i][1]<small)

{

small=temp[i][1];

j=i;

}

}

return j;

}

int check\_visited(int k)

{

int i;

for(i=0;i<l;i++)

{

if(visited[i]==k)

return 1;

}

return 0;

}

void display\_path()

{

int i;

printf("\n\nThe Path of tour is:\n\n");

for(i=0;i<n;i++)

printf("\t%d",visited[i]+1);

printf("\t1");

printf("\n\nMinimum cost of tour is:");

printf(" %d\n",cost[l-1]);

}

int reduced\_matrix(int b[10][10])

{

int i,j,reduced\_sum=0,min;

for(i=0;i<n;i++) //row minimization

{

min=find\_row\_min(b,i);

reduced\_sum+=min;

for(j=0;j<n;j++)

{

if(b[i][j]!=99)

b[i][j]-=min;

}

}

for(i=0;i<n;i++) // column minimization

{

min=find\_col\_min(b,i);

if(min!=0){

reduced\_sum+=min;

for(j=0;j<n;j++)

{

if(b[j][i]!=99)

b[j][i]-=min;

}

}

}

printf("\nReduced Matrix for node no:%d is :\n",node\_no++);

for(i=0;i<n;i++)

{

printf("\n");

for(j=0;j<n;j++)

{

printf("\t%d",b[i][j]);

}

}

return reduced\_sum;

}

void tspbb()

{

int temp[10][2],q,p,m[10][10],sum=0;

int i,j,k;

while(l<n)

{

q=0;

for(i=0;i<n;i++)

{

p=check\_visited(i);

if(p==0)

temp[q++][0]=i;

}

for(i=0;i<q;i++)

{

for(k=0;k<n;k++) //copy full reduced matrix a to matrix m

{

for(j=0;j<n;j++)

{

m[k][j]=a[k][j];

}

}

for(k=0;k<l;k++)

{

for(j=0;j<n;j++)

{

m[visited[k]][j]=99; // make row infinity for visited city

}

}

for(k=1;k<l;k++)

{

for(j=0;j<n;j++)

{

m[j][visited[k]]=99; // make col infinity for next visited city

}

}

for(j=0;j<n;j++) // make col infinity for the city to be visited

{

m[j][temp[i][0]]=99;

}

for(j=0;j<n;j++) //make m[j][i] infinity for i to j path

{

if(visited[j+1]!=-1)

m[visited[j+1]][0]=99;

}

m[temp[i][0]][0]=99;

sum=reduced\_matrix(m);

temp[i][1]=cost[l-1] + sum + a[visited[l-1]][temp[i][0]];

printf("\ncost of node no %d =%d",node\_no-1,temp[i][1]);

}//for loop i

p=min1(temp,q); //find minimum cost node as E-node

visited[l]=temp[p][0];

cost[l++]=temp[p][1];

printf("\n\n minimum cost city is =%d and its cost is :%d",temp[p][0]+1,temp[p][1]);

}//while loop

}

int main()

{

int i,j;

printf("\nEnter No. of Cities: ");

scanf("%d",&n);

printf("\nEnter Cost Matrix:\n");

for( i=0;i<n;i++)

{

printf("\n Enter Elements of Row # : %d\n",i+1);

for( j=0;j<n;j++)

{

scanf("%d",&a[i][j]);

}

visited[i]=-1; // initially visited array is set to -1

}

printf("\n\nThe Given cost matrix is:\n\n");

for( i=0;i<n;i++)

{

printf("\n");

for( j=0;j<n;j++)

printf("\t%d",a[i][j]);

}

visited[l]=0;

cost[l++]=reduced\_matrix(a);

tspbb();

display\_path();

return 0;

}

Enter No. of Cities: 5

Enter Cost Matrix:

Enter Elements of Row # : 1

99 20 30 10 11

Enter Elements of Row # : 2

15 99 16 4 2

Enter Elements of Row # : 3

3 5 99 2 4

Enter Elements of Row # : 4

19 6 18 99 3

Enter Elements of Row # : 5

16 4 7 16 99

The Given cost matrix is:

99 20 30 10 11

15 99 16 4 2

3 5 99 2 4

19 6 18 99 3

16 4 7 16 99

Reduced Matrix for node no:1 is :

99 10 17 0 1

12 99 11 2 0

0 3 99 0 2

15 3 12 99 0

11 0 0 12 99

Reduced Matrix for node no:2 is :

99 99 99 99 99

99 99 11 2 0

0 99 99 0 2

15 99 12 99 0

11 99 0 12 99

cost of node no 2 =35

Reduced Matrix for node no:3 is :

99 99 99 99 99

1 99 99 2 0

99 3 99 0 2

4 3 99 99 0

0 0 99 12 99

cost of node no 3 =53

Reduced Matrix for node no:4 is :

99 99 99 99 99

12 99 11 99 0

0 3 99 99 2

99 3 12 99 0

11 0 0 99 99

cost of node no 4 =25

Reduced Matrix for node no:5 is :

99 99 99 99 99

10 99 9 0 99

0 3 99 0 99

12 0 9 99 99

99 0 0 12 99

cost of node no 5 =31

minimum cost city is =4 and its cost is :25

Reduced Matrix for node no:6 is :

99 99 99 99 99

99 99 11 99 0

0 99 99 99 2

99 99 99 99 99

11 99 0 99 99

cost of node no 6 =28

Reduced Matrix for node no:7 is :

99 99 99 99 99

1 99 99 99 0

99 1 99 99 0

99 99 99 99 99

0 0 99 99 99

cost of node no 7 =50

Reduced Matrix for node no:8 is :

99 99 99 99 99

1 99 0 99 99

0 3 99 99 99

99 99 99 99 99

99 0 0 99 99

cost of node no 8 =36

minimum cost city is =2 and its cost is :28

Reduced Matrix for node no:9 is :

99 99 99 99 99

99 99 99 99 99

99 99 99 99 0

99 99 99 99 99

0 99 99 99 99

cost of node no 9 =52

Reduced Matrix for node no:10 is :

99 99 99 99 99

99 99 99 99 99

0 99 99 99 99

99 99 99 99 99

99 99 0 99 99

cost of node no 10 =28

minimum cost city is =5 and its cost is :28

Reduced Matrix for node no:11 is :

99 99 99 99 99

99 99 99 99 99

99 99 99 99 99

99 99 99 99 99

99 99 99 99 99

cost of node no 11 =28

minimum cost city is =3 and its cost is :28

The Path of tour is:

1 4 2 5 3 1

Minimum cost of tour is: 28